In the UML sequence diagram, messages are illustrated with an arrow symbol, together with the name of the message and its parameters (if present). In this way, in UML a message is categorically divided into two parts:

The name of the message specifies the *event*.

The arguments of the message contain the *information* that is attached to the message, so that the receiver can perform the necessary activities. Control information belongs in this category as well.

We refer to the information that is exchanged as a *business object*, if the information:

Is coherent

Is structured

Covers the requirements for a certain activity (e.g., invoice, passenger list)

Is self-contained (no in-house reference keys, etc.)

Outlives individual interactions

In the UML model of system integration, business objects are structured information sent as arguments in a message from a sender to a receiver.

Creating

System Operation Contracts

• Identify each system operation in a system sequence diagram.

• Write the responsibilities of that operation in the contract.

• Write the preconditions in terms of the required changes in the domain model.

• Add the post conditions(changes expected after operation) and exceptions.

Design principle: cohesive (packages in java are cohesive)

Generalisation, specialisation association is the Java extends(inheritance)

Realisation is the java implements for interfaces

Use case model shows functionality we want – Use case is the WHAT

Domain model (conceptual model) shows us important concepts and relationships. See how business operates with respect to the concepts within. Domain Driven design. Use language that describes that business. Patient stays as patient etc. IS the HOW

Words are HUGELY important

Class diagram contains classes and relationships between those classes

They can be: association, generalisation/specialisation or aggregation

Something has a something, Faculty has a department is modelled using aggregation. (hollow diamond)

Is made up of, modelled using composition (black diamond)

Moving into design model, we flesh out the classes by adding operations and associations.

How do we work out what operations? Responsibility driven design. What responsibilities do you want a class to have?

2 types:

Knowing responsibilities - A class should know its attributes and its associated classes.

Doing responsibility -

Classes collaborate to find out information like cost of visit.

For a clinic to determine the cost of a visit for a patient, what would the clinic be given to do that?

Who the patient is (name)

Which visit (date)

EXAM question:

Interaction sequence diagram showing interactions between objects.

Given this, create class diagram that would support this sequence diagram.

# Systems Analysis

Primary Goal:

To state accurately the users’ requirements for a new information processing system.

Secondary Goals:

1. To understand the users’ requirements.

2. To communicate the current understanding of the proposed system

3. To prevent expensive mistakes

4. To state a design problem

5. To state the conditions for system acceptance

* Step 1. Identify the business events and make an event table.
* Step 2. Identify the use cases and produce a use case diagram for the system.
* Step 3. Write a use case narrative describing the system’s response to each business event.
* Step 4. Draw a system sequence diagram for each use case scenario.
* Step 5. Produce a domain model showing the concepts, attributes, and associations in the problem domain of the system.
* Step 6. Write a contract for each system operation.

A six-step process for object-oriented analysis is introduced . Together, these six steps produce an event model, a use case model, system sequence diagrams, a model of the problem domain, and system operation contracts.

An event is an occurrence which takes place at a specific time and triggers a predetermined response from the system.

Event analysis identifies the events to which the system is expected to respond, names the inputs and outputs, and identifies the actors.

An event is an occurrence which takes place at a specific time and initiates or triggers a redetermined response from the system.

An external event is an event which occurs outside the system boundary.

An internal event is an event which occurs inside the system boundary.

A temporal event is an event which occurs at a prespecified time.

# Procedure for Object-Oriented Systems Analysis

Step 1. Identify the business events and make an event table.

Step 2. Identify the use cases and produce a use case diagram for the system.

Step 3. Write a use case narrative describing the system’s response to each business event.

Step 4. Draw a system sequence diagram for each use case scenario.

Step 5. Produce a domain model showing the concepts, attributes and associations in the problem domain of the system.

Step 6. Write a contract for each system operation.

A concept is an abstraction of a thing, a person, or an idea. It is represented by a rectangle.

An attribute is a characteristic of a concept which may have a value. Attribute names appear in the lower compartment of the concept rectangle.

An association is a significant connection between concepts. It is represented by a line connecting a pair of concepts.

# Domain Modelling

A Domain Model (DM) illustrates conceptual classes in the problem domain. We illustrate a DM using the class diagrams of the UML. It shows:

* Domain objects or conceptual classes
* Associations between conceptual classes
* Attributes of conceptual classes.

Advantages of using a Domain Model

• Once the domain concepts have been modeled, the model becomes a stable basis for subsequent development of applications in the domain.

• The concepts of the conceptual model can be mapped into physical design or implementation constructs .

• The Conceptual /Domain model describes and constrains the scope of the problem domain.

• The domain model can be effectively used to verify and validate the understanding of the problem domain among various stakeholders.

• It is especially helpful as a communication tool and a focusing point both amongst the different members of the business team as well as between the technical and business teams.

Domain Model

•Identify noun & noun phrases in text of the scenario for the Fully Dressed Use Case and consider them as conceptual classes or attributes.

•Use Noun & Noun phrase first then concept category list.

•From both noun & noun phrases and concept category list, we identify candidate concepts, some may actually be attributes of concepts.